IN THE CLAIMS:

Please add new claims 25-48 as shown below. This listing of claims will replace all prior versions, and listings, of claims in the present application:

1. (Original) A method of transmitting executable software from a server to a client computer, the method comprising:

segmenting each of a plurality of applications into a collection of executable blocks;

forming an InitBlock Bundle comprising blocks executable during initialization of the plurality of applications, at least one block from each application being included in the InitBlock Bundle;

sending the InitBlock Bundle to a client computer; and

sending other blocks from the plurality of collections of executable blocks to the client computer subsequent to a start of execution of the InitBlock Bundle.

2. (Original) The method of claim 1 wherein:

the plurality of applications comprise at least one application subscribed to by a user and at least one application not subscribed to by the user; and

the method further comprises:

monitoring execution of applications subscribed to by the user to determine an application usage pattern; and

based on the usage pattern, sending data to the client terminal to display information about a first one of the unsubscribed applications.

3. (Original) The method of claim 2 wherein:

the data to display information about one of the unsubscribed applications comprises an offer to subscribe to the first one of the unsubscribed applications.

4. (Original) The method of claim 3 further comprising:

in response to the offer to subscribe, receiving data at the server indicating acceptance of the offer; and

sending data to the client terminal to enable execution of the first one of the unsubscribed applications by the user.

5. (Original) The method of claim 4 wherein:

the InitBlock Bundle comprises access control data; and

sending data to enable execution comprises sending changed access control data from the server to the client computer.

6. (Original) The method of claim 5 wherein sending the changed access control data comprises automatically sending in response to a subscription request received at the server from the client computer.

7. (Original) The method of claim 5 wherein the access control data comprises an encryption key enabling access to blocks of subscribed-to applications.

8. (Original) The method of claim 1 further comprising:

from each of a plurality of service providers, sending to a client terminal an InitBlock Bundle comprising a plurality of initialization blocks;

monitoring execution of blocks in each of said InitBlock Bundles to determine a usage pattern; and

forming a new InitBlock Bundle based on the usage pattern.

9. (Original) The method of claim 8 wherein the new InitBlock Bundle comprises executable blocks associated with applications from different ones of the service providers.

10. (Original) The method of claim 1 wherein at least one of the blocks in the InitBlock Bundle is a shared block executable during the initialization phase of different ones of the applications.

11. (Original) The method of claim 1 wherein the InitBlock Bundle comprises a set of blocks sufficient to enable execution of each of the plurality of applications to a point when the application awaits user input.

12. (Original) The method of claim 1 wherein forming the InitBlock Bundle comprises: monitoring usage of a plurality of different applications; and wherein forming the initialization block comprises forming based on the monitored usage.

13. (Original) The method of claim 1 further comprising:

sending from the server to the client a plurality of key values, the key values identifying ones of the collection of collections of executable blocks;

receiving a response at the server from the client indicating blocks identified by the key values that are already stored at the client; and wherein

sending the InitBlock Bundle comprises omitting blocks stored already stored at the client.

14. (Original) A method of streaming data from a server to a client computer, the method comprising:

sending from a server to a client a key value identifying a streamable block;

receiving a response at the server from the client indicating whether the client has a locally stored copy of the block; and

sending the block to the client if the client does not have a locally stored copy.

15. (Original) The method of claim 14 wherein:

sending a key value further comprises sending a group of other key values identifying other streamable blocks;

receiving a response further comprises receiving a response indicating whether the client has locally stored copies of ones of the other blocks; and

sending the block further comprises sending ones of the other blocks that are not locally stored at the client.

16. (Original) The method of claim 15 further comprising:

at the client, storing first data associating key values with locally stored blocks; and

processing the first data to determine whether the client has a locally stored copy of a block identified by the received key value.

17. (Original) The method of claim 14 wherein the key value is computed at the server using a hashing algorithm.

18. (Original) The method of claim 17 wherein the hashing algorithm comprises a digital signature algorithm.

19. (Original) A computer system comprising:

a database storing a plurality of executable applications segmented into a plurality of code blocks, each application's plurality of code blocks comprising a set of initialization code blocks;

a processor operatively coupled to a network interface, to the database and to a computer readable data storage media comprising instructions to configure the processor to:

form an initialization block comprising initialization code blocks for at least two of the plurality of applications; and

send the initialization block to a client computer operatively coupled to the network interface.

20. (Original) The system of claim 19 wherein the data storage media further comprises instructions to configure the processor to:

monitoring execution of initialization code blocks at the client computer to determine a usage pattern; and

forming a new InitBlock Bundle based on the usage pattern.

21. (Original) The system of claim 19 wherein:

the system further comprises a database comprising a plurality of user profiles, each user profile comprising security data to control usage of ones of the plurality of applications by a respective user;

the data storage media further comprises instructions to query the database of user profiles to access security data associated with a first user;

instructions to process the security data to determine application restriction data associated with the first user; and

instructions to send the application restriction data to the first client computer.

22. (Original) The system of claim 21 wherein the application restriction data comprises further comprises data preventing user access to the unsubscribed second application.

23. (Original) A computer readable data storage apparatus storing instructions for configuring a computer to:

send to a client terminal a key value identifying a streamable block;

receive a response from the client terminal indicating whether the client terminal has a locally stored copy of the block; and

send the block to the client terminal if the client does not have a locally stored copy.

24. (Original) The apparatus of claim 23 wherein:

the instructions to send a key value further comprises instructions to simultaneously send a group of other key values identifying other streamable blocks;

the instructions to receive a response further comprises instructions to receive a response indicating whether the client terminal has locally stored copies of ones of the other blocks; and

the instructions to send the block further comprises instructions to send ones of the other blocks that are not locally stored at the client.

25. (New) A method comprising:

dividing at least a portion of a software application into blocks;

packaging the blocks into a repository from which the blocks can be individually extracted; and

generating an application package that includes at least the repository, the application package for use by a streaming server to stream the software application to a client.

26. (New) The method of claim 25, further comprising compressing each block prior to said packaging.

27. (New) The method of claim 25 wherein each block has a size equal to a code page size used during file reads by an operating system expected to be present on a system executing the software application.

28. (New) The method of claim 25, wherein the block size is four kilobytes.

29. (New) The method of claim 25, further comprising:

generating an index to the repository; and

including the index in the application package.

30. (New) The method of claim 25, wherein the application has a file structure, the method further comprising including a specification of the file structure of the application in the application package.

31. (New) The method of claim 25, wherein the repository is configured to permit access to blocks with reference to a source application file and an offset in the source application file.

32. (New) The method of claim 25, further comprising:

determining environmental changes introduced to a computer system by installation of the software application on the computer system; and

providing an environmental install package comprising data indicating the determined environmental changes for distribution.

33. (New) The method of claim 32, wherein said providing the environmental install package comprises including the environmental install package in the application package.

34. (New) The method of claim 25, further comprising:

determining a startup set comprising at least a minimal set of blocks sufficient for execution of the software application to be initiated; and

providing a startup streamlet set for distribution.

35. (New) The method of claim 34, wherein said providing the startup set comprises including the startup streamlet set in the application package.

36. (New) The method of claim 34, wherein said determining the startup set comprises:

executing the software application;

monitoring file block load requests at least until a designated startup point has been reached by the software application; and

identifying file blocks which were loaded prior to reaching the startup point.

37. (New) The method of claim 36, wherein the startup point comprises a point at which the application waits for user input.

38. (New) The method of claim 25, further comprising:

generating a predictive model for use in determining likely blocks to be loaded by the application when in a given state; and

providing the predictive model for distribution.

39. (New) The method of claim 38, wherein said providing the predictive model comprises including the predictive model in the application package.

40. (New) A method comprising:

determining a file structure of a software application;

dividing a file of the software application into blocks;

packaging the blocks into a repository from which the blocks can be individually extracted; and

creating an application package which includes the repository and a specification of the file structure, the application package for use by a streaming server to stream the software application to a target system.

41. (New) The method of claim 40 wherein each of the blocks has a size equal to a code page size used during file reads by an operating system expected to be present on a system executing the application.

42. (New) The method of claim 40, wherein the repository is configured to permit access to blocks with reference to a source application file and an offset in the source application file.

43. (New) The method of claim 40, further comprising:

determining environmental changes introduced to a computer system by installation of the software application on the computer system;

providing an environmental install package comprising data indicating the determined environmental changes for distribution; and

including the environmental install package in the application package.

44. (New) The method of claim 40, further comprising:

determining at least a minimal set of blocks sufficient for execution of the software application to be initiated; and

including said minimal set of blocks as a startup streamlet set in the application package.

45. (New) The method of claim 44, wherein said determining at least a minimal set of blocks sufficient for execution of the application to be initiated comprises:

executing the application;

monitoring file block load requests at least until a designated startup point has been reached by the application; and

identifying file blocks which were loaded prior to reaching the startup point.

46. (New) The method of claim 45, wherein the startup point comprises a point when the application waits for user input.

47. (New) The method of claim 40, further comprising:

generating a predictive model for use in determining likely blocks to be loaded by the application when in a given state; and

providing the predictive model for distribution.

48. (New) The method of claim 47, wherein said providing the predictive model comprises including the predictive model in the application package.